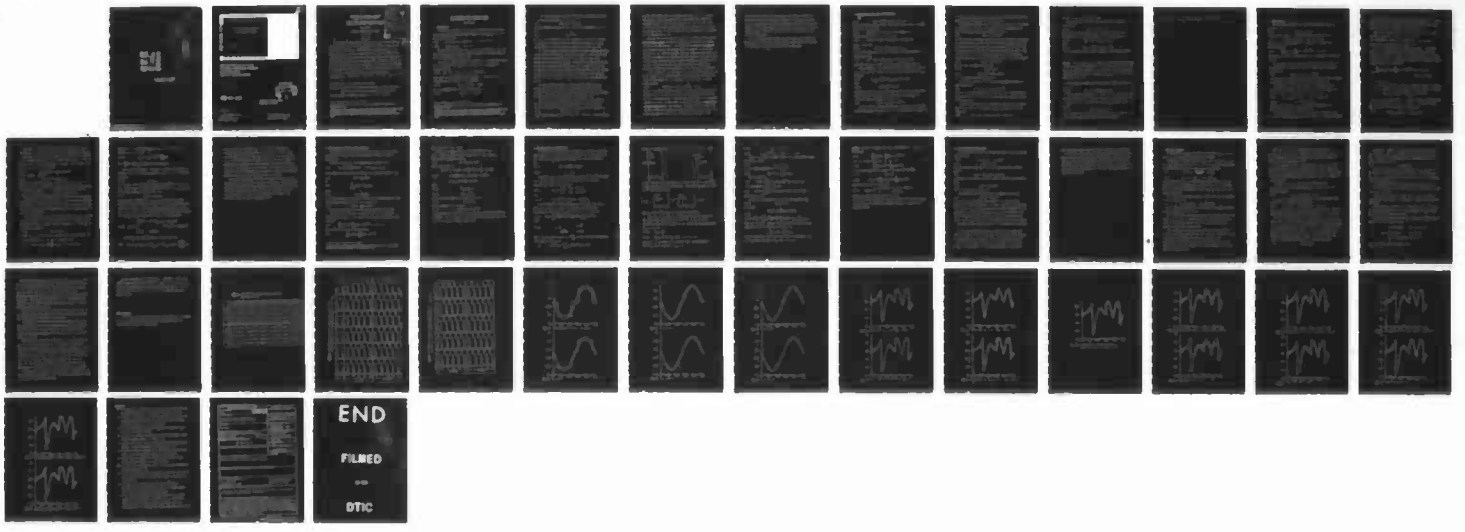
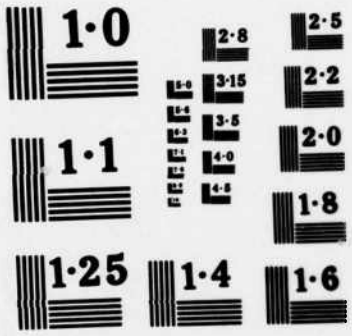
AD-A158 131 ON HIGH-ORDER ACCURATE INTERPOLATION FOR NON-OSCILLATORY SHOCK CAPTURING SCHEMES(U) WISCONSIN UNIV-MADISON MATHEMATICS RESEARCH CENTER A HARTEN

1/1

UNCLASSIFIED JUN 85 MRC-TSR-2829 DAAG29-82-C-0090 F/G 12/1 NL





MRC Technical Summary Report # 2829

ON HIGH-ORDER ACCURATE INTERPOLATION FOR NON-OSCILLATORY SHOCK CAPTURING SCHEMES

Ami Harten

Mathematics Research Center
University of Wisconsin—Madison
610 Walnut Street
Madison, Wisconsin 53705

June 1985

(Received June 6, 1985)

Approved for public release
Distribution unlimited

Sponsored by

U. S. Army Research Office
P. O. Box 12211
Research Triangle Park
North Carolina 27709

National Aeronautics and
Space Administration
Washington, DC 20546

(A)

UNIVERSITY OF WISCONSIN - MADISON
MATHEMATICS RESEARCH CENTER

ON HIGH-ORDER ACCURATE INTERPOLATION FOR NON-OSCILLATORY SHOCK CAPTURING SCHEMES

Ami Harten*

Technical Summary Report #2829

June 1985

Abstract

In this paper we describe high-order accurate Godunov-type schemes for the computation of weak solutions of hyperbolic conservation laws that are essentially non-oscillatory. It is shown that the problem of designing such schemes reduces to a problem in approximation of functions, namely that of reconstructing a piecewise smooth function from its given cell averages to high order accuracy and without introducing large spurious oscillations. To solve this reconstruction problem we introduce a new interpolation technique that when applied to piecewise smooth data gives high-order accuracy wherever the function is smooth but avoids having a Gibbs-phenomenon at discontinuities.

Additional keywords: variation; operators (mathematics)

AMS (MOS) Subject Classification: 65M05, 65D05

Key Words: Conservation laws, Godunov-type schemes, high-order accuracy, non-oscillatory interpolation, Gibbs-phenomenon, cell-average.

Work Unit Number 3 - Numerical Analysis and Scientific Computing

---

* School of Mathematical Sciences, Tel-Aviv University and Department of Mathematics, UCLA.

Sponsored by the United States Army under Contract No. DAAG29-80-C-0041 while in residence at the MRC, University of Wisconsin-Madison, and by NASA Consortium Agreement #NCA2-IR390-403 and ARO Grant #DAAG29-82-0090 while at UCLA.

ON HIGH-ORDER ACCURATE INTERPOLATION FOR NON-OSCILLATORY SHOCK CAPTURING SCHEMES

Ami Harten*

1. Introduction.

In this paper we present an interpolation technique that gives rise to new high-order accurate non-oscillatory schemes for the numerical solution of scalar conservation laws:

$$u_t + f(u)_x \equiv u_t + a(u)u_x = 0 \quad , \quad -\infty < x < \infty, \; t > 0 \tag{1.1a}$$

$$u(x,0) = u_0(x) \quad , \quad -\infty < x < \infty \quad . \tag{1.1b}$$

We assume the initial data $u_0(x)$ to be piecewise-smooth functions that are either periodic or of compact support, and denote the evolution operator of the unique entropy solution by $E(t)$.

Let $v_h(x,t)$ denote a numerical approximation to a weak solution $u(x,t)$ of (1.1), such that $v_j^n = v_h(x_j,t_n)$, $x_j = jh$, $t_n = n\tau$, satisfies the conservation form

$$v_j^{n+1} = v_j^n - \lambda\left(\hat{f}_{j+1/2} - \hat{f}_{j-1/2}\right) \equiv [E_h(\tau)\cdot v^n]_j \quad . \tag{1.2a}$$

Here $E_h(\tau)$ is the numerical solution operator, $\lambda = \tau/h$ and $\hat{f}_{j+1/2}$, the numerical flux, is a function of $2k$ variables

$$\hat{f}_{j+1/2} = \hat{f}(v^n_{j-k+1},\ldots,v^n_{j+k}) \tag{1.2b}$$

which is consistent with (1.1a) in the sense that

$$\hat{f}(u,u,\ldots,u) = f(u) \quad . \tag{1.2c}$$

---

* School of Mathematical Sciences, Tel-Aviv University and Department of Mathematics, UCLA.

Sponsored by the United States Army under Contract No. DAAG29-80-C-0041 while in residence at the MRC, University of Wisconsin-Madison, and by NASA Consortium Agreement #NCA2-IR390-403 and ARO Grant #DAAG29-82-0090 while at UCLA.

It is well known that if the total variation of the numerical solution (1.2) is uniformly bounded in h for $0 \leq t \leq T$

$$TV(v_h(\cdot,t)) \leq C \cdot TV(u_0) \quad , \tag{1.3}$$

then any refinement sequence $h \to 0$, $\tau = O(h)$, has a subsequence $h_j \to 0$ so that

$$v_{h_j} \xrightarrow{L_1} u \tag{1.4}$$

where u is a weak solution of (1.1). Furthermore, if all limit solutions (1.4) satisfy an extropy condition that implies uniqueness of the initial value problem (1.1), then the numerical scheme is convergent (see [4]).

Our goal in designing numerical schemes is to obtain a computer code that is both reliable and efficient. To achieve reliability we would like to use schemes that are total-variation-stable in the sense of (1.3); to get efficiency we need high-order accuracy. Unfortunately it became clear in the development of shock-capturing schemes that it is not easy to satisfy both requirements at the same time: Endowing a scheme with a property that implies automatic control over the growth of the total variation of the numerical solution may very well lead to a restriction on the order of accuracy of the scheme.

The first successful attempt to achieve nonlinear stability was to require positivity of the numerical solution operator, which led to the development of monotone schemes. However the requirement of positivity automatically implies first order accuracy of the scheme (see [2]).

The next step in the development was to consider the larger class of total-variation-diminishing (TVD) schemes, where the numerical solution operator is required to diminish the total variation of any BV function v

$$TV(E_h(\tau) \cdot v) \leq TV(v) \quad ; \tag{1.5}$$

these schemes trivially satisfy (1.3) with $C = 1$ (see [3]). We were able to

construct TVD schemes that in the sense of local truncation error are high-order accurate everywhere except at local extrema where they necessarily degenerate into first order accuracy, (see [9], [3], [1], [7], [8]). The perpetual damping of local extrema determines the cumulative global error in the L_p-norm to be of $(1 + \frac{1}{p})$-th order, i.e. only first-order in the maximum norm but second-order in L_1.

Recently ([5]) we went one step further and introduced a larger class of non-oscillatory schemes, in which the application of the numerical solution operator to any mesh function v is required not to increase the number of local extrema (note that this statement does not depend on h nor on the smoothness of v). Unlike TVD schemes, which are a subset of this class, non-oscillatory schemes are not required to damp the values of each local extremum at every single time-step, but are allowed to occasionally accentuate a local extremum. Because of this last property we were able to construct non-oscillatory schemes that are second-order accurate also in the maximum norm.

In the present paper we relax the control over the possible growth of the total variation of the numerical solution even further and consequently are able to design schemes that are accurate to any finite order r. These schemes satisfy

$$TV(E_h(\tau)\cdot u) \leq TV(u) + O(h^{r+1}), \qquad u \in U \tag{1.6}$$

where U is the set of functions that are C_∞ except at a finite number of points (in any finite interval) where they may have a discontinuity in the function or some derivative (describes the generic form of the "computable" solutions of (1.1)). We shall refer to this class of schemes as essentially non-oscillatory schemes. The inequality (1.6) ensures that the scheme does not have a Gibbs-like phenomenon of oscillations that are proportional to the size of the jump at a discontinuity; the permissible increase in total

variation is solely due to the smooth part of the function. Unlike the second-order accurate non-oscillatory schemes of [5] which are a subset of this class, essentially non-oscillatory schemes may occasionally increase the number of local extrema. However these extra oscillations are on the level of truncation error in the smooth part and therefore can be regarded as "non-essential oscillations".

In the following we shall present high-order Godunov-type schemes that satisfy (1.6), and show that the design problem reduces to solving a problem in interpolation; the latter is the main topic of this paper.

2. High-order accurate Godunov-type schemes.

Let

$$\bar{u}(x) = \frac{1}{h}\int_{-h/2}^{h/2} u(x+y)dy \equiv (A_h \cdot u)(x) \tag{2.1}$$

denote the sliding average of $u(x)$. The sliding average $\bar{u}(x,t)$ of a weak solution of (1.1) satisfies

$$\frac{\partial}{\partial t}\bar{u}(x,t) + \frac{1}{h}\left[f(u(x+\tfrac{h}{2},t)) - f(u(x-\tfrac{h}{2},t))\right] = 0 \quad . \tag{2.2a}$$

Integrating (2.2a) from t to $t+\tau$ we get

$$\bar{u}(x,t+\tau) = \bar{u}(x,t) - \lambda\left[\hat{f}(x+\tfrac{h}{2},t;u) - \hat{f}(x-\tfrac{h}{2},t;u)\right] \tag{2.2b}$$

where $\lambda = \tau/h$ and

$$\hat{f}(x,t;u) = \frac{1}{h}\int_0^\tau f(u(x,t+\eta))d\eta \quad . \tag{2.2c}$$

Thus the exact weak solution of (1.1) satisfies the following relation on the computational mesh

$$\bar{u}_j^{n+1} = \bar{u}_j^{n} - \lambda\left[\hat{f}(x_{j+1/2},t_n;u) - \hat{f}(x_{j-1/2},t_n;u)\right] \quad ; \tag{2.3}$$

where $\bar{u}_j^{n} = \bar{u}(x_j,t_n)$ is the cell average of the solution at time t_n.

Next we compare (2.3) to the numerical scheme in *conservation form* (1.2) where we set $v_j^n \equiv \bar{u}_j^n$. We see that if the resulting numerical flux $\hat{f}_{j+1/2}$ satisfies

$$\hat{f}_{j+1/2} = \frac{1}{\tau}\int_0^\tau f(u(x_{j+1/2},t_n+\eta))d\eta + O(h^r) \tag{2.4a}$$

then

$$v_j^{n+1} = \bar{u}_j^{n+1} + O(h^{r+1}) \quad , \tag{2.4b}$$

provided that the coefficient in the $O(h^r)$ term in (2.4a) is sufficiently smooth. Relation (2.4b) shows that in the sense of cell-averages the truncation error of the scheme is $O(h^{r+1})$, i.e.

$$\bar{u}(t+\tau) - E_h(\tau)\cdot\bar{u}(t) = O(h^{r+1}) \quad . \tag{2.5}$$

We shall refer to a scheme that satisfies (2.4)-(2.5) as an r-th order Godunov-type scheme. (Note the difference from Lax-Wendroff-type schemes that

are derived by approximating a Taylor expansion of the solution and where the truncation error is made small in a pointwise sense.)

We observe that although (2.3) is a relation between the cell averages of the solution at t_n and t_{n+1}, the evaluation of the numerical flux in (2.4a) involves point values of the solution. Since

$$\bar{u}(x) - u(x) = O(h^2) \tag{2.6}$$

we have to devise a technique to recover point values from given cell averages to a desired accuracy, in order to obtain Godunov-type schemes that are more than second-order accurate. The rest of this paper is devoted to describing an algorithm for the solution of this problem in approximation: Given $\{\bar{u}_j\}$, cell averages of $u \in U$ (i.e. piecewise smooth with a finite number of discontinuities) find $R(x;\bar{u})$ that reconstructs $u(x)$ to any finite order r

$$R(x;\bar{u}) = u(x) + O(h^r) \tag{2.7a}$$

wherever $u(x)$ is smooth, and such that

$$\bar{R}(x_j;\bar{u}) = \bar{u}_j \quad , \tag{2.7b}$$

$$TV(R(\cdot;\bar{u})) \leq TV(u) + O(h^r) \quad ; \tag{2.7c}$$

here $\bar{R}$ denotes the sliding average of R.

It is easy to see that once we solve this approximation problem, the Godunov-type scheme

$$v^{n+1} \equiv E_h(\tau)\cdot v^n = A_h \cdot E(\tau) \cdot R(\cdot;v^n) \quad , \tag{2.8}$$

where A_h is the cell-averaging operator (2.1) and $E(\tau)$ is the evolution operator of (1.1), is r-th order accurate and essentially non-oscillatory in the sense of (1.6).

To see that the scheme (2.8) is essentially non-oscillatory we observe that both A_h and $E(\tau)$ are positive operators and therefore also TVD. Consequently

$$TV(E_h(\tau)\cdot\bar{u}) \equiv TV(A_h \cdot E(\tau) \cdot R(\cdot;\bar{u})) \leq TV(R(\cdot;\bar{u})) \tag{2.9}$$

and (1.6) follows immediately from (2.7c).

Next we show that (2.8) is r-th order accurate in the sense of (2.4). Let us denote

(2.10a) $$v_n(\cdot,t) = E(t)\cdot R(\cdot;v^n) \quad .$$

Using (2.7b) and the fact that $v_n(x,t)$ is an exact solution of the conservation law (1.1a) we can rewrite (2.8) in the form (2.3), i.e.

(2.10b) $$v_j^{n+1} = v_j^n - \lambda\left(\hat{f}_{j+1/2} - \hat{f}_{j-1/2}\right)$$

where

(2.10c) $$\hat{f}_{j+1/2} = \frac{1}{\tau}\int_0^\tau f\left(v_n\left(x_{j+1/2},t\right)\right)dt \quad .$$

(2.4a) then follows from (2.7a) and the stability of entropy solutions of (1.1).

Remarks: (1) The scheme (2.8) is the abstract form of Godunov-type schemes. In practice we use an approximation to (2.10) which is obtained by using an appropriate numerical quadrature for the integral in (2.10c), and using an approximate solution operator to evaluate $v_n\left(x_{j+1/2},t\right)$ in (2.10a). For more details see [5] and [6].

(2) We initialize the computation by taking cell-averages of the given initial data, i.e.

(2.11a) $$v_j^0 = \bar{u}_j^0 = \frac{1}{h}\int_{-h/2}^{h/2} u_0(x_j+y)dy \quad .$$

When we apply the scheme(2.8) N times, $N\cdot\tau = t$, and the initial data are such that the solution is smooth, we expect the truncation error (2.5) to accumulate linearly. Thus at the end of the time loop we get (since $\tau = O(h)$)

(2.11b) $$v_j^N = \bar{u}(x_j,t) + O(h^r) \quad .$$

Since in general we are interested in pointwise values we output $R(x_j;v^N)$ which gives us the pointwise data to the desired accuracy:

(2.11c) $$R(x;v^N) = R(x;\bar{u}) + O(h^r) = u(x,t) + O(h^r) \quad .$$

3. Interpolation.

In this section we present a new interpolation technique $H_m(x;u)$ for functions $u \in U$

$$H_m(x_j;u) = u(x_j) \quad . \tag{3.1a}$$

The interpolant $H_m(x;u)$ is a piecewise polynomial function of x, i.e.

$$H_m(x;u) = q_{m,j+1/2}(x;u) \quad \text{for} \quad x_j \leq x \leq x_{j+1} \tag{3.1b}$$

where $q_{m,j+1/2}$ is a polynomial in x of degree m. Wherever $u(x)$ is smooth we get that

$$\frac{d^k}{dx^k} H_m(x;u) = \frac{d^k}{dx^k} u(x) + O(h^{m+1-k}) \quad , \quad 0 \leq k \leq m \quad . \tag{3.2}$$

(We use here the standard convention that $k = 0$ corresponds to the function itself.)

The new feature of this interpolation technique is that, although u may be discontinuous, the interpolant $H_m(x;u)$ is essentially non-oscillatory in the sense that

$$TV(H_m(\cdot;u)) \leq TV(u) + O(h^{m+1}) \quad . \tag{3.3}$$

This interpolation technique will be used in the following sections to develop an algorithm for the solution of the reconstruction problem (2.7).

To accomplish (3.2) we take $q_{m,j+1/2}(x;u)$ to be the m-th degree polynomial that interpolates $u(x)$ at the $m + 1$ successive points $\{x_i\}$, $i_m(j) \leq i \leq i_m(j) + m$, that include x_j and x_{j+1}, i.e.

$$q_{m,j+1/2}(x_i;u) = u(x_i) \quad , \quad i_m(j) \leq i \leq i_m(j) + m \quad , \tag{3.4a}$$

$$1 - m \leq i_m(j) - j \leq 0 \quad . \tag{3.4b}$$

Clearly there are exactly m such polynomials corresponding to the m different choices of $i_m(j)$ subject to (3.4b).

We have assumed that $u(x)$ has a finite number of discontinuities. Hence for h sufficiently small there are at least $m + 1$ points of smooth-

ness between any two successive discontinuities. Consequently if (x_j,x_{j+1}) is an interval in which $u(x)$ is smooth, there is at least one choice of $i_m(j)$ such that $u(x)$ is smooth in $x_{i_m(j)} \le x \le x_{i_m(j)+m}$.

Next we show that any algorithm to assign $i_m(j)$ (3.4) to (x_j,x_{j+1}) that has the property:

(3.5) smoothness in (x_j,x_{j+1}) ==> smoothness in $(x_{i_m(j)},x_{i_m(j)+m})$,

yields $H_m(x;u)$ that satisfies both (3.2) and (3.3).

Let x_0 be a point that has a neighborhood in which u is smooth. Hence for h sufficiently small there is an interval (x_j,x_{j+1}) that includes x_0 in which u is smooth. It follows then from (3.5), (3.4) and the fact that $q_{m,j+1/2}$ is an interpolating polynomial that uses data from an interval in which u is smooth that

$$\frac{d^k}{dx^k} q_{m,j+1/2}(x;u) = \frac{d^k}{dx^k} u(x) + O(h^{m+1-k}) \quad \text{for} \quad 0 \le k \le m \tag{3.6}$$

$$\text{and} \quad x_j < x < x_{j+1} \ ;$$

this implies (3.2).

We turn now to show (3.3). First let us consider an interval (x_j,x_{j+1}) in which u is smooth. It follows immediately from (3.6) with $k = 0$ that

$$TV_{[x_j,x_{j+1}]}(q_{m,j+1/2}) \le TV_{[x_j,x_{j+1}]}(u) + O(h^{m+1}) \ . \tag{3.7}$$

Moreover in the intervals in which u is smooth and $\frac{du}{dx}$ is bounded away from zero, it follows from (3.6) with $k = 1$ that for h sufficiently small $\frac{d}{dx} q_{m,j+1/2} \neq 0$ in such an interval and consequently

$$TV_{[x_j,x_{j+1}]}(q_{m,j+1/2}) = TV_{[x_j,x_{j+1}]}(u) \ . \tag{3.8}$$

Next let us consider an interval (x_j,x_{j+1}) that contains a single discontinuity of u. It turns out that for h sufficiently small $q_{m,j+1/2}$, for any choice of $i_m(j)$ in (3.4), is monotone in (x_j,x_{j+1}); this implies (3.8). To simplify the argument let us consider the case that u is a step-function with the discontinuity located in (x_j,x_{j+1}). In this case $u(x_i) = u(x_{i+1})$ for all $i \neq j$ and therefore

(3.9a) $$q_{m,j+1/2}(x_i;u) = q_{m,j+1/2}(x_{i+1};u)$$

for all indices i such that

(3.9b) $$i \neq j \quad \text{and} \quad 0 \leq i-i_m(j) \leq m-1 \ .$$

From (3.9a) it follows that $\frac{d}{dx} q_{m,j+1/2}$ has a root in each of the $m-1$ intervals (x_i,x_{i+1}) with indices i satisfying (3.9b). Since $\frac{d}{dx} q_{m,j+1/2}$ is a polynomial of degree $m-1$ it cannot have an additional root in (x_j,x_{j+1}) and therefore $q_{m,j+1/2}$ is strictly monotone there.

We conclude from the above analysis that an increase in total variation is possible only in those intervals in which u has a smooth local extremum. Since there is only a finite number of such intervals and since the increase there (3.7) is on the level of interpolation error, (3.3) follows (see [6] for a more detailed analysis).

In the following we present an algorithm to assign $i_m(j)$ to (x_j,x_{j+1}). In order to satisfy (3.5) this algorithm makes use of the information about smoothness contained in a table of divided differences of u. The latter can be defined recursively by

(3.10a) $$u[x_i] = u(x_i)$$

(3.10b) $$u[x_i,\ldots,x_{i+k}] = (u[x_{i+1},\ldots,x_{i+k}] - u[x_i,\ldots,x_{i+k-1}])/(x_{i+k}-x_i) \ .$$

It is well known that if u is C_∞ in $[x_i,x_{i+k}]$ then

(3.11a) $$u[x_i,\ldots,x_{i+k}] = \frac{1}{k!} \frac{d^k}{dx^k} u(\xi_{i,k}) \ , \quad x_i < \xi_{i,k} < x_{i+k} \ .$$

However if u has a jump discontinuity in the p-th derivative in this interval, $0 < p < k$, then

$$u[x_i,\ldots,x_{i+k}] = O\left(h^{-k+p}\left[\frac{d^p}{dx^p}u\right]\right) \quad ; \tag{3.11b}$$

here $\left[\frac{d^p}{dx^p}u\right]$ denotes the jump in the p-th derivative.

Our algorithm is recursive: It arrives at $H_r(x;u)$, which amounts to assigning $i_r(j)$, by successively evaluating $i_k(j)$, $k = 1,\ldots,r$. We start by setting

$$i_1(j) = j \quad , \tag{3.12a}$$

i.e. $q_{1,j+1/2}$ is the first degree polynomial interpolating u at x_j and x_{j+1}. Let us assume that we have already defined $i_k(j)$, i.e. $q_{k,j+1/2}$ is the k-th degree polynomial interpolating u at

$$x_{i_k(j)},\ldots,x_{i_k(j)+k} \quad . \tag{3.13}$$

We consider now as candidates for $q_{k+1,j+1/2}$ the two (k+1)-th degree interpolating polynomials obtained by adding to (3.13) the neighboring point to the left or the one to the right; this corresponds to setting $i_{k+1}(j) = i_k(j) - 1$ or $i_{k+1}(j) = i_k(j)$, respectively. We choose the one that gives a (k+1)-th order divided difference that is smaller in absolute value:

$$i_{k+1}(j) = \begin{cases} i_k(j) - 1 & \text{if } \left|u[x_{i_k(j)-1},\ldots,x_{i_k(j)+k}]\right| \\ & \qquad < \left|u[x_{i_k(j)},\ldots,x_{i_k(j)+k+1}]\right| \\ i_k(j) & \text{otherwise} \end{cases} . \tag{3.12b}$$

Using Newton's form of interpolation it is easy to see that

$$q_{k+1,j+1/2} - q_{k,j+1/2} = u[x_{i_{k+1}(j)},\ldots,x_{i_{k+1}(j)+k+1}] \cdot \prod_{i=i_k(j)}^{i_k(j)+k} (x-x_i) \quad . \tag{3.14}$$

Since the product in the RHS of (3.14) is the same for both choices in (3.12b), we see that our algorithm selects as $q_{k+1,j+1/2}$ this (k+1)-th polynomial that deviates the least from $q_{k,j+1/2}$ in (x_j,x_{j+1}). Clearly if h is sufficiently small it follows from (3.11) that the algorithm (3.12) satisfies the requirement (3.5), and consequently has the desired properties (3.2) and (3.3). However if h is not small, (3.14) shows that the algorithm attempts to find the "least oscillatory" polynomial (subject to the restricted choice in (3.12b)), and thus is meaningful also in this case.

In the next two sections we describe two different techniques to solve the reconstruction problem (2.7) in terms of interpolation. The importance of using the particular interpolation described in this section is that its non-oscillatory nature goes over to the reconstruction algorithm. The non-oscillatory nature of $R(x;\bar{u})$ (2.7c) is demonstrated in section 7 by numerical examples (analysis is presented in [6]).

4. Reconstruction via the primitive function.

In this section we apply a technique frequently used in area-preserving approximations[1] in order to solve the reconstruction problem in terms of interpolation.

Given cell-averages $\bar{u}_j$ of a piecewise-smooth function $u \in U$

$$\bar{u}_j = \frac{1}{h_j} \int_{x_{j-1/2}}^{x_{j+1/2}} u(y)dy \quad , \quad h_j = x_{j+1/2} - x_{j-1/2} \tag{4.1}$$

we can immediately evaluate the point-values of the primitive function $U(x)$

$$U(x) = \int_{x_0}^{x} u(y)dy \tag{4.2}$$

by

$$\sum_{i=i_0}^{j} h_j \bar{u}_j = U\left(x_{j+1/2}\right) \quad . \tag{4.3}$$

Since

$$u(x) \equiv \frac{d}{dx} U(x) \tag{4.4}$$

we can apply interpolation to the point values of the primitive function (4.3) and then obtain an approximation to $u(x)$ by defining

$$R(x;\bar{u}) = \frac{d}{dx} H_r(x;U) \quad . \tag{4.5}$$

We note that this procedure does not require uniformity of the mesh.

The primitive function $U(x)$ is smoother than $u(x)$ (by one extra derivative) and therefore $U \in U$. Hence we get from (3.2) that wherever $u(x)$ is smooth

$$\frac{d^k}{dx^k} H_r(x;U) = \frac{d^k}{dx^k} U(x) + O(h^{r+1-k}) \quad , \quad 0 \leq k \leq r \quad . \tag{4.6}$$

Using (4.4) and (4.5) we can rewrite (4.6) as

$$\frac{d^\ell}{dx^\ell} R(x;\bar{u}) = \frac{d^\ell}{dx^\ell} u(x) + O(h^{r-\ell}) \quad , \tag{4.7}$$

---

[1] I thank Nira Dyn from Tel-Aviv University for pointing this out. A similar approach was taken by Woodward [10] and Zalesak [11].

which implies (2.7a) for $\ell = 0$.

We turn now to study $\bar{R}(x;\bar{u})$, the sliding average of (4.5):

$$(4.8)\qquad \bar{R}(x;\bar{u}) = \frac{1}{h}\int_{-h/2}^{h/2} R(x+y)dy = \frac{1}{h}\left[H_r(x+\tfrac{h}{2};\, U) - H_r(x-\tfrac{h}{2};\, U)\right] .$$

Denoting the interpolation error by

$$(4.9)\qquad e(x) = H_r(x;U) - U(x)$$

we can rewrite (4.8) as

$$\bar{R}(x;\bar{u}) = \frac{1}{h}\left[U(x+\tfrac{h}{2}) - U(x-\tfrac{h}{2})\right] + \frac{1}{h}\left[e(x+\tfrac{h}{2}) - e(x-\tfrac{h}{2})\right]$$

$$(4.10)\qquad = \frac{1}{h}\int_{x-h/2}^{x+h/2} u(y)dy + \frac{1}{h}\left[e(x+\tfrac{h}{2}) - e(x-\tfrac{h}{2})\right]$$

$$= \bar{u}(x) + \frac{1}{h}\left[e(x+\tfrac{h}{2}) - e(x-\tfrac{h}{2})\right] .$$

Since

$$(4.11a)\qquad e(x_{j+1/2}) = 0 ,$$

$$(4.11b)\qquad e(x) = O(h^{r+1}) ,$$

we get from (4.10) that (4.5) satisfies (2.7b), i.e.

$$(4.12a)\qquad \bar{R}(x_j;\bar{u}) = \bar{u}_j$$

and that wherever $e(x)$ is smooth

$$(4.12b)\qquad \bar{R}(x;\bar{u}) = \bar{u}(x) + O(h^{r+1}) .$$

It follows from (4.12) that $\bar{R}(x;\bar{u})$ is a piecewise-polynomial interpolation of degree r of $\bar{u}$. Note that (4.12b) is one order more accurate than (4.7) with $\ell = 0$, which is $R(x;\bar{u}) = u(x) + O(h^r)$.

5. Reconstruction via deconvolution.

In this section we describe another technique to reconstruct a piecewise smooth function $u \in U$ from its given cell-averages $\bar{u}_j$. We assume that the mesh is uniform and consider the cell-averages $\bar{u}_j$ to be point values of $\bar{u}(x)$,

$$\bar{u}(x) = \frac{1}{h}\int_{-h/2}^{h/2} u(x-y)dy \tag{5.1a}$$

$$\bar{u}_j = \bar{u}(x_j) \quad . \tag{5.1b}$$

The function $\bar{u}(x)$ (5.1a) was referred to earlier as the sliding average of u. Here we consider it to be the convolution of $u(x)$ with the characteristic function of a cell $\psi_h(x)$

$$\psi_h(x) = \begin{cases} 1/h & \text{for} \quad |x| < h/2 \\ 0 & \text{for} \quad |x| > h/2 \end{cases} , \tag{5.2a}$$

i.e.

$$\bar{u}(x) = \int_{-\infty}^{\infty} u(x-y)\psi_h(y)dy = (u*\psi_h)(x) \quad . \tag{5.2b}$$

In the following we describe a procedure to reconstruct $u(x)$ up to $O(h^r)$ for any finite r; this will be referred to as "finite-order deconvolution". Expanding $u(x-y)$ in (5.2b) around $y = 0$, we get

$$\bar{u}(x) = \sum_{k=0}^{\infty} \frac{u^{(k)}(x)}{k!} \frac{(-1)^k}{h} \int_{-h/2}^{h/2} y^k dy \equiv \sum_{k=0}^{\infty} \alpha_k h^k u^{(k)}(x) \tag{5.3a}$$

where

$$\alpha_k = \begin{cases} 0 & k \text{ odd} \\ \dfrac{2^{-k}}{(k+1)!} & k \text{ even} \end{cases} . \tag{5.3b}$$

Multiplying both sides of (5.3a) by $h^\ell \dfrac{d^\ell}{dx^\ell}$ and then truncating the expansion in the RHS at $O(h^r)$ we get

$$h^\ell \bar{u}^{(\ell)}(x) = \sum_{k=0}^{r-\ell-1} \alpha_k h^{k+\ell} u^{(k+\ell)}(x) + O(h^r) \quad . \tag{5.4a}$$

Writing the relations (5.4a) for $\ell = 0,\ldots,r-1$ in matrix form, we obtain

$$(5.4b)\quad \begin{bmatrix} \bar{u}(x) \\ h\bar{u}'(x) \\ h^2\bar{u}''(x) \\ \vdots \\ h^{r-1}\bar{u}^{(r-1)}(x) \end{bmatrix} = \begin{pmatrix} 1 & 0 & \alpha_2 & 0 & \alpha_4 & \cdots & \alpha_{r-1} \\ & \ddots & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & \ddots & \alpha_4 \\ & & & & \ddots & \ddots & 0 \\ & & 0 & & & \ddots & \alpha_2 \\ & & & & & \ddots & 0 \\ & & & & & & 1 \end{pmatrix} \begin{bmatrix} u(x) \\ hu'(x) \\ h^2u''(x) \\ \vdots \\ h^{r-1}u^{(r-1)}(x) \end{bmatrix} + O(h^r) \quad .$$

Let us denote the coefficient matrix in the RHS of (5.4b) by C. This matrix is upper triangular and diagonally dominant. Multiplying both sides of (5.4b) by C^{-1} from the left we get

$$(5.4c)\quad \begin{pmatrix} u(x) \\ hu'(x) \\ \vdots \\ h^{r-1}u^{(r-1)}(x) \end{pmatrix} = C^{-1} \begin{pmatrix} \bar{u}(x) \\ h\bar{u}'(x) \\ \vdots \\ h^{r-1}\bar{u}^{(r-1)}(x) \end{pmatrix} + O(h^r) \quad .$$

Relation (5.4c) is the essence of finite-order deconvolution.

Given $\bar{u}_j$ (5.1) we apply the interpolation technique of section 3 to $\bar{u}$ and compute $O(h^r)$ approximations to $h^\ell \bar{u}^{(\ell)}(x)$ at x_j by taking appropriate derivatives of $H_m(x;\bar{u})$, $m \geq r-1$; denote these approximations by $\bar{D}_{\ell,j}$, $0 \leq \ell \leq r-1$:

$$(5.5a)\quad \bar{D}_{0,j} = \bar{u}_j$$

$$(5.5b)\quad \bar{D}_{\ell,j} = h^\ell \bar{u}^{(\ell)}(x_j) + O(h^r) \quad , \quad \ell = 1,\ldots,r-1 \quad .$$

Using $\bar{D}_{\ell,j}$ in the RHS of (5.4c) we obtain $O(h^r)$ approximations to $h^\ell u^{(\ell)}(x_j)$ which we denote by $D_{\ell,j}$, i.e.

(5.6a) $$D_{0,j} = u(x_j) + O(h^r)$$

(5.6b) $$D_{\ell,j} = h^\ell u^{(\ell)}(x_j) + O(h^r) \quad , \quad 1 \le \ell \le r-1 \quad .$$

Since C is an upper triangular matrix, $D_{\ell,j}$ are computed by back-substitution: We set

(5.7a) $$D_{r-1,j} = \bar{D}_{r-1,j}$$

and then compute backwards for $k = r-2,\ldots,0$

(5.7b) $$D_{k,j} = \bar{D}_{k,j} - \sum_{\ell=k+1}^{r-1} \alpha_\ell D_{\ell,j} \quad .$$

Finally we define

(5.8) $$R(x;\bar{u}) = \sum_{k=0}^{r-1} \frac{1}{k!} D_{k,j}[(x-x_j)/h]^k \quad \text{for} \quad |x-x_j| < h/2 \quad .$$

It follows immediately from (5.6) that

(5.9) $$R(x;\bar{u}) = \sum_{k=0}^{r-1} \frac{1}{k!} u^{(k)}(x_j)(x-x_j)^k + O(h^r) = u(x) + O(h^r) \quad ,$$

which implies (2.7a).

To see that the reconstruction (5.8) satisfies property (2.7b)

(5.10a) $$\bar{R}(x_j;\bar{u}) = \bar{u}_j$$

we evaluate

$$\bar{R}(x_j;\bar{u}) = \frac{1}{h} \int_{-h/2}^{h/2} R(x_j - y;\bar{u})dy$$

to get

(5.10b) $$\bar{R}(x_j;\bar{u}) = \sum_{k=0}^{r-1} \frac{1}{k!} \frac{D_{k,j}}{h^k} \frac{(-1)^k}{h} \int_{-h/2}^{h/1} y^k dy = D_{0,j} + \sum_{k=1}^{r-1} \alpha_k D_{k,j} \quad ,$$

where α_k are (5.3b). Relation (5.10a) then follows immediately from comparing the RHS of (5.10b) to (5.7b) with $k = 0$

$$\bar{D}_{0,j} = D_{0,j} + \sum_{k=1}^{r-1} \alpha_k D_{k,j}$$

and then using (5.5a). We note that this result is independent of the particular values assigned to $\bar{D}_{\ell,j}$ for $1 \le \ell \le r-1$.

Remarks: (1) $\bar{u}(x)$ is smoother than $u(x)$, and therefore $u \in U \Rightarrow \bar{u} \in U$. Consequently

$$\frac{d^\ell}{dx^\ell} H_m(x;\bar{u}) = \frac{d^\ell}{dx^\ell} \bar{u}(x) + O(h^{m+1-k}) \quad . \tag{5.11a}$$

Since H_m is only continuous at x_j, for $k \geq 1$

$$\frac{d^k}{dx^k} H_m(x_j+0;\bar{u}) \neq \frac{d^k}{dx^k} H_m(x_j-0;\bar{u}) \quad . \tag{5.11b}$$

In the numerical experiments presented in this paper we have taken

$$\bar{D}_{0,j} = \bar{u}_j \tag{5.12a}$$

$$\bar{D}_{k,j} = h^k M\left(\frac{d^k}{dx^k} H_m(x_j+0;\bar{u}), \frac{d^k}{dx^k} H_m(x_j-0;\bar{u})\right) \tag{5.12b}$$

where $M(x,y)$ is the min mod function

$$M(x,y) = \begin{cases} s \cdot \min(|x|,|y|) & \text{if } \operatorname{sgn}(x) = \operatorname{sgn}(y) = s \\ 0 & \text{otherwise} \end{cases} \quad . \tag{5.13}$$

Clearly this choice satisfies (5.5) for $m \geq r-1$.

(2) The finite order deconvolution extends very easily to the case where the mollifier $\psi_h(y)$ is replaced by a smoother function or a function with a different support.

6. The constant coefficient case.

In this section we consider the Godunov-type scheme (2.8) in the constant coefficient case

(6.1a) $$u_t + au_x = 0, \qquad a = \text{const.}$$

(6.1b) $$u(x,0) = u_0(x) \quad .$$

The exact solution in this case is just pure translation with a constant speed a

(6.2) $$u(x,t) = [E(t)\cdot u_0](x) = u_0(x-at) \quad .$$

Consequently we can express the numerical scheme explicitly in the following form:

(6.3) $$v_j^{n+1} = \bar{R}(x_j - a\tau; v^n) \quad ;$$

here $\bar{R}$ is the sliding average of R.

The truncation error in the sense of cell-averages (2.5) is then

(6.4) $$\bar{u}(x-a\tau) - \bar{R}(x-a\tau;\bar{u}) \quad ,$$

which measures how well $\bar{R}(\cdot;\bar{u})$ approximates $\bar{u}$. We have noted that generally if $R(\cdot;\bar{u})$ approximates u to $O(h^r)$, then $\bar{R}(\cdot;\bar{u})$ approximates $\bar{u}$ to $O(h^{r+1})$, which corresponds to an r-th order accurate scheme. To have this gain of one extra order of accuracy we need spacial smoothness of the reconstruction error; this is evident from (2.4a) and more directly from (4.10) - (4.11). When we do not have this smoothness we find that the scheme is r-th order accurate in the L_1-norm but not in the maximum norm (see section 7).

We note that although the problem to be solved (6.1) is linear, the numerical scheme (6.3) is highly nonlinear. The nonlinearity enters through the interpolation, where the stencil (3.12) is chosen differently at each point and each time level, depending on local smoothness of the numerical solution. In this respect (6.3) is conceptually different from standard

finite-difference schemes where the stencil is arbitrarily predetermined.

We would also like to point out that this scheme breaks aways from the somewhat artificial notion of "upstream differencing": The decision what stencil to use in the reconstruction is made on the basis of smoothness considerations and has nothing to do with the "direction of the wind". The latter enters only when applying $E(\tau)$ in (2.8). The resulting stencil is a combination of the two.

7. Numerical examples.

In this section we present some numerical examples in order to illustrate the nature of the approximations described in this paper.

The first set of examples deals with smooth data. In Figs. 1a to 1f we show approximations to $u(x) = \sin \pi x$, $-1 \le x \le 1$; these were reconstructed via deconvolution (see section 5) from the cell-averages input

$$\bar{u}_j = \bar{u}(x_j) = \frac{\sin(\pi h/2)}{(\pi h/2)} \cdot \sin(\pi x_j) \tag{7.1a}$$

$$x_j = -1 + j \cdot \frac{2}{N} \quad , \quad 0 \le j \le N-1 \quad . \tag{7.1b}$$

In this example we took $N = 6$ and extended the data outside $[-1,1]$ by periodicity. The cell-averages input is shown in Figs. 1a - 1f by circles. Both the reconstructed approximation $R(x;\bar{u})$ (5.8) and $u(x) = \sin \pi x$ are shown by solid lines.

Let us denote the polynomial degree of $H(x;\bar{u})$ (5.12) by P_I and that of $R(x;\bar{u})$ (5.8) by P_R:

$$P_I = \deg[H(x;\bar{u})] \quad , \quad P_R = \deg[R(x;\bar{u})] \quad . \tag{7.2}$$

Fig. 1a shows the reconstruction associated with the original first-order accurate Godunov scheme: $P_I = P_R = 0$.

Fig. 1b shows the reconstruction associated with the "second-order" TVD schemes [3]: $P_I = P_R = 1$. Comparing it to Fig. 1a we see that the local extrema are flattened in exactly the same way - this demonstrates the degeneracy to first-order accuracy there.

Fig. 1c shows the reconstruction associated with the second-order accurate non-oscillatory scheme of [5]: $P_I = 2$, $P_R = 1$. Comparing it to Fig. 1b we observe a considerable improvement in the quality of the approximation. We also note that the reconstruction increases the total variation of the input data; however, this increase is small - it is of the size of the truncation error.

Figs. 1d, 1e and 1f show the reconstruction corresponding to $P_I = k$, $P_R = k-1$ for $k = 3,4$ and 5, respectively. We observe that unlike the previous approximations, the reconstruction here does not go through the circles; this is a consequence of (2.6).

The reconstruction $R(x;\bar{u})$ is discontinuous at $x_{j+1/2}$; the size of the jump is proportional to the reconstruction error. As we increase the accuracy of the reconstruction going from Fig. 1a to Fig. 1f, we observe that the size of the jumps becomes smaller and smaller; in Fig. 1f the jumps are hardly noticeable.

In Table 1 we list the L_∞-norm of the interpolation error $|H(x;\bar{u}) - \bar{u}(x)|$ and of the reconstruction error $|R(x;\bar{u}) - u(x)|$ for a refinement sequence

(7.3) $$N = 4,\ 8,\ 16,\ 32,\ 64 \quad \text{in (7.1)} .$$

We turn now to examine the performance of the resulting Godunov-type scheme (2.8) in the constant coefficient case

(7.4) $$u_t + u_x = 0, \quad u(x,0) = \sin \pi x \ , \quad -1 \le x \le 1 \ .$$

We input the initial data in the form of the cell-averages (7.1). Again we assume periodicity in space, which implies a period of 2 in time as well.

In Table 2 we list both the L_∞- and the L_1-error at $t = 2$ of the scheme (2.10), (6.3) where $R(x;\bar{u})$ is reconstruction via deconvolution with P_I and P_R defined by (7.2). The calculations in this table were performed for the refinement sequence (7.3) with $\lambda = \tau/h = 0.8$. In addition to the actual error we also list the quantity r which is the computational order of accuracy. Assuming the error to be exactly $e_h = \text{const} \cdot h^r$ we evaluate r for any two successive calculations in the refinement sequence by

(7.5) $$r = \log_2(e_h/e_{h/2}) \ .$$

In Table 3 we repeat the calculations in Table 2 for the scheme (2.10), (6.3) where now $R(x;\bar{u})$ is reconstruction via the primitive function (4.5). Here P_U denotes the polynomial degree of the interpolation $H(x;U)$ of the primitive function (4.2), i.e.

$$P_U = \deg[H(x;U)] \quad . \tag{7.6}$$

We observe from Table 2 that the schemes with $P_I = k$ and $P_R = k-1$ are (at least) k-th order accurate in both L_∞ and L_1. The TVD scheme with $P_I = P_R = 1$ is a little bit better than first-order in L_∞ and a little bit worse than second-order in L_1.

We observe from Table 3 that the schemes with the odd order $P_U = 3$ and $P_U = 5$ seem to be third and fifth-order accurate, respectively, in both the L_∞ and L_1 norms.

However when P_U is an even number, $P_U = 2, 4, 6,$ this order is realized only in the L_1-sense. In the L_∞-norm it seems to be one order lower in accuracy.

Comparing Table 2 and Table 3 for the fine mesh $N = 64$, we find that the scheme from Table 2 with $P_I = k$ and the scheme with $P_U = k+1$ from Table 3 seem to give comparable accuracy.

We turn now to examine the performance of the various approximations when applied to the discontinuous function

$$u(x) = \begin{cases} -x \sin(3\pi x^2/2) & \text{for } -1 < x < -\frac{1}{3} \\ |\sin(2\pi x)| & \text{for } |x| < \frac{1}{3} \\ 2x - 1 - \frac{1}{6}\sin(3\pi x) & \text{for } \frac{1}{3} < x < 1 \end{cases} , \tag{7.7}$$

which we extend periodically outside $[-1,1]$.

The circles in both Figs. 2a and 2b show the given cell-average of (7.7) at $N = 40$ uniformly spaced mesh points. The two solid lines in Figure 2a are $\bar{u}(x)$ and $H(x;\bar{u})$ with $P_I = 6$. The two solid lines in Fig. 2b are $u(x)$ and $R(x;\bar{u})$ with $P_R = 5$. We see in both figures that the approximations give good accuracy in the smooth part and are essentially non-oscillatory. This is as to be expected since $N = 40$ provides sufficient resolution of the problem in the sense that there are at least 7 points of smoothness inbetween discontinuities.

In Figs. 3a and 3b we repeat the calculation in Fig. 2 for $N = 15$. It is interesting to note that in spite of the lack of resolution in this case both approximations are essentially non-oscillatory.

To compare the two reconstruction techniques we repeat in Fig. 3c the same case described in Fig. 3b, but now using reconstruction via primitive function with $P_U = 6$. The main difference is in the rounding of local extrema at discontinuities present in Fig. 3b; this is due to the min mod operation in (5.12b).

Next we apply the Godunov-type scheme (2.10), (6.3) to the solution of $u_t + u_x = 0$ with the initial data (7.7). As before we initialize the computation by taking cell-averages of the initial data and use $\lambda = \tau/h = 0.8$. In Figs. 4, 5 and 6 we present calculations performed with reconstruction via deconvolution using $P_I = P_R = 1$ in Fig. 4; $P_I = 2$, $P_R = 1$ in Fig. 5 and $P_I = 4$, $P_R = 3$ in Fig. 6. The results are presented at: (a) $t = 2$, (b) $t = 4$. The circles in these figures show the reconstructed numerical approximation at the mesh points; the solid line shows the exact solution.

We observe that in all cases we get non-oscillatory approximations, and that the schemes are dissipative in time. The quality of the results improves with increasing formal order of accuracy.

In Fig. 7 we compare the two reconstruction techniques. In Fig. 7b we repeat the same calculation as in Figs. 4 to 6 using $P_I = 5$ and $P_R = 4$. In Fig. 7a we use reconstruction via primitive function with $P_U = 6$ in (7.6). Unlike the situation in Figs. 3b - 3c, here the two schemes produce very similar results.

Acknowledgement.

I would like to thank Sukumar Chakravarthy, Bjorn Engquist and Stan Osher for many stimulating discussions and for various contributions to this research (which is part of a joint project).

Table 1. Interpolation and reconstruction L_∞-errors for $u(x) = \sin \pi x$.

	N	$P_I=1, P_R=1$	$P_I=2, P_R=1$	$P_I=3, P_R=2$	$P_I=4, P_R=3$	$P_I=5, P_R=4$	$P_I=6, P_R=5$
$\|\|H(\cdot;\bar{u}) - u\|\|_\infty$ Interpolation Error	4	1.86×10^{-1}	1.86×10^{-1}	7.39×10^{-2}	7.39×10^{-2}	3.17×10^{-2}	3.17×10^{-2}
	8	6.85×10^{-2}	2.84×10^{-2}	7.63×10^{-3}	3.16×10^{-3}	9.37×10^{-4}	3.88×10^{-4}
	16	1.87×10^{-2}	3.72×10^{-3}	5.36×10^{-4}	1.07×10^{-4}	1.70×10^{-5}	3.39×10^{-6}
	32	4.78×10^{-3}	4.71×10^{-4}	3.45×10^{-5}	3.40×10^{-6}	2.76×10^{-7}	2.72×10^{-8}
	64	1.20×10^{-3}	5.91×10^{-5}	2.17×10^{-6}	1.07×10^{-6}	4.36×10^{-9}	2.14×10^{-10}
$\|\|R(\cdot;\bar{u}) - u\|\|_\infty$ Reconstruction Error	4	2.57×10^{-1}	2.57×10^{-1}	1.07×10^{-1}	1.07×10^{-1}	4.69×10^{-2}	4.69×10^{-2}
	8	1.64×10^{-1}	6.28×10^{-2}	1.52×10^{-2}	5.34×10^{-3}	1.74×10^{-3}	5.85×10^{-4}
	16	4.87×10^{-2}	1.37×10^{-2}	1.13×10^{-3}	2.15×10^{-4}	3.28×10^{-5}	6.22×10^{-6}
	32	1.27×10^{-2}	3.27×10^{-3}	7.54×10^{-5}	7.18×10^{-6}	5.37×10^{-7}	5.23×10^{-8}
	64	3.20×10^{-3}	8.07×10^{-4}	7.88×10^{-6}	2.52×10^{-7}	8.49×10^{-9}	4.16×10^{-10}

Table 2. Solution of $u_t + u_x = 0$, $u(x,0) = \sin \pi x$ at $t = 2$ using reconstruction via deconvolution with $\lambda = \tau/h = 0.8$.

	N	$P_I = 1$ $P_R = 1$	r	$P_I = 2$ $P_R = 1$	r	$P_I = 3$ $P_R = 2$	r	$P_I = 4$ $P_R = 3$	r	$P_I = 5$ $P_R = 4$	r	$P_I = 6$ $P_R = 5$	r
L_∞-ERROR	4	5.62×10^{-1}		4.23×10^{-1}		2.88×10^{-1}		1.82×10^{-1}		1.20×10^{-1}		7.92×10^{-2}	
			1.12		2.27		3.10		4.52		4.78		6.37
	8	2.59×10^{-1}		8.77×10^{-2}		3.35×10^{-2}		7.95×10^{-3}		4.38×10^{-3}		9.56×10^{-4}	
			1.21		2.45		2.95		4.86		4.92		6.87
	16	1.12×10^{-1}		1.60×10^{-2}		4.34×10^{-3}		2.74×10^{-4}		1.45×10^{-4}		8.19×10^{-6}	
			1.24		2.19		3.14		4.78		5.11		6.92
	32	4.74×10^{-2}		3.51×10^{-3}		4.92×10^{-4}		9.95×10^{-6}		4.21×10^{-6}		6.74×10^{-8}	
			1.27		2.14		3.16		4.20		5.13		6.38
	64	1.97×10^{-2}		7.99×10^{-4}		5.52×10^{-5}		5.42×10^{-7}		1.20×10^{-7}		8.08×10^{-10}	
L_1-ERROR	4	6.57×10^{-1}		4.61×10^{-1}		3.31×10^{-1}		2.28×10^{-1}		1.51×10^{-1}		1.03×10^{-1}	
			1.47		2.07		3.48		4.52		5.44		6.48
	8	2.37×10^{-1}		1.10×10^{-1}		2.97×10^{-2}		9.97×10^{-3}		3.47×10^{-3}		1.15×10^{-3}	
			1.34		2.36		3.28		4.66		5.10		6.67
	16	9.39×10^{-2}		2.14×10^{-2}		3.05×10^{-3}		3.95×10^{-4}		1.01×10^{-4}		1.13×10^{-5}	
			1.67		2.26		3.91		4.78		5.79		6.92
	32	2.95×10^{-2}		4.47×10^{-3}		2.03×10^{-4}		1.44×10^{-5}		1.83×10^{-6}		9.33×10^{-8}	
			1.82		2.13		3.82		4.72		5.85		6.89
	64	8.37×10^{-3}		1.02×10^{-3}		1.44×10^{-5}		5.46×10^{-7}		3.18×10^{-8}		7.85×10^{-10}	

Table 3. Solution of $u_t + u_x = 0$, $u(x,0) = \sin \pi x$ at $t = 2$, using reconstruction via primitive function and $\lambda = \tau/h = 0.8$.

	N	$P_U = 2$	r	$P_U = 3$	r	$P_U = 4$	r	$P_U = 5$		$P_U = 6$	r
L_∞-ERROR	4	4.877×10^{-1}		3.193×10^{-1}		2.391×10^{-1}		1.535×10^{-1}		1.103×10^{-1}	
			0.99		2.61		2.96		4.52		4.78
	8	2.442×10^{-1}		5.221×10^{-2}		3.072×10^{-2}		6.664×10^{-3}		4.026×10^{-3}	
			1.20		2.84		2.92		4.91		4.84
	16	1.065×10^{-1}		7.310×10^{-3}		4.050×10^{-3}		2.223×10^{-4}		1.401×10^{-4}	
			1.20		2.96		3.14		4.97		4.95
	32	4.636×10^{-2}		9.384×10^{-4}		4.593×10^{-4}		7.112×10^{-6}		4.083×10^{-6}	
			1.25		3.00		3.16		5.00		5.12
	64	1.952×10^{-2}		1.171×10^{-4}		5.132×10^{-5}		2.228×10^{-7}		1.177×10^{-7}	
L_1-ERROR	4	8.233×10^{-1}		5.353×10^{-1}		3.665×10^{-1}		2.328×10^{-1}		1.620×10^{-1}	
			1.79		2.96		3.71		4.82		5.39
	8	2.373×10^{-1}		6.870×10^{-2}		2.800×10^{-2}		8.267×10^{-3}		3.864×10^{-3}	
			1.37		2.98		3.10		4.92		5.20
	16	9.176×10^{-2}		8.731×10^{-3}		3.271×10^{-3}		2.729×10^{-4}		1.053×10^{-4}	
			1.60		2.99		3.88		4.98		5.82
	32	3.024×10^{-2}		1.096×10^{-3}		2.225×10^{-4}		8.631×10^{-6}		1.864×10^{-6}	
			1.83		3.00		3.82		5.00		5.88
	64	8.506×10^{-3}		1.368×10^{-4}		1.582×10^{-5}		2.701×10^{-7}		3.164×10^{-8}	

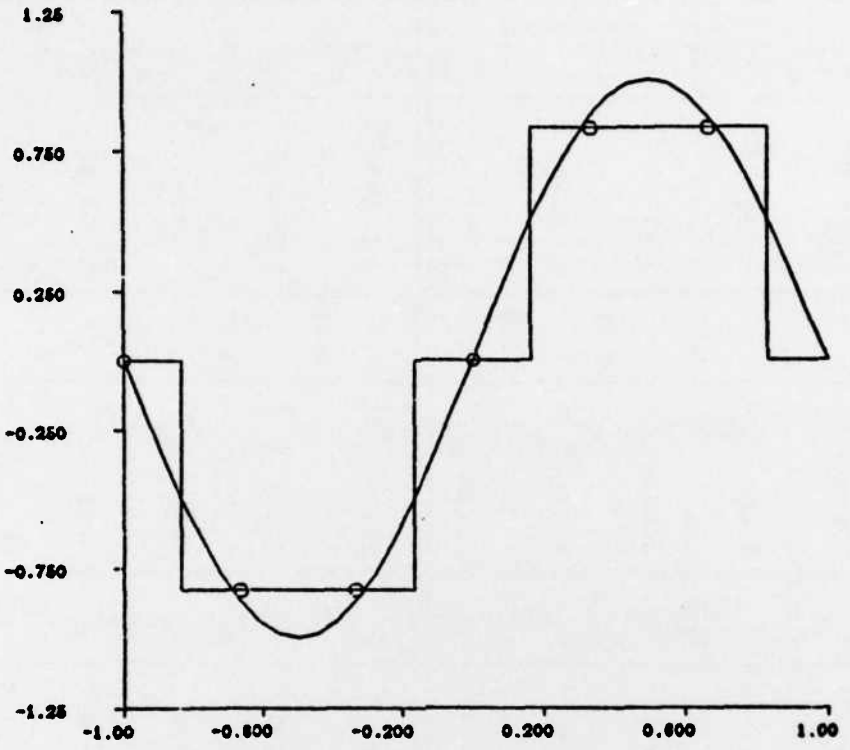


Fig. 1a. Reconstruction of sin πx. $P_I = P_R = 0$.

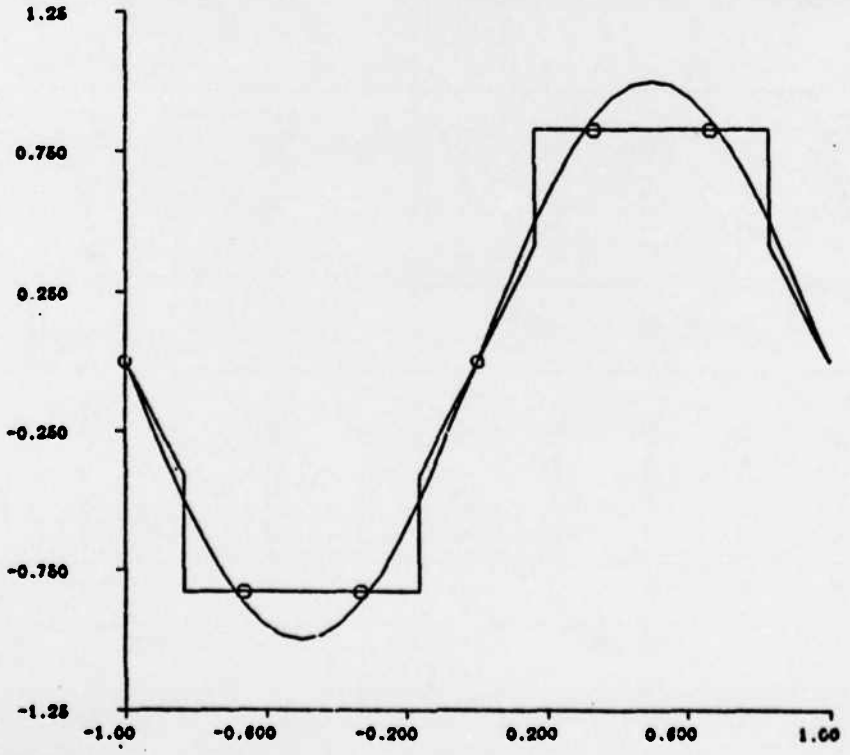


Fig. 1b. $P_I = P_R = 1$.

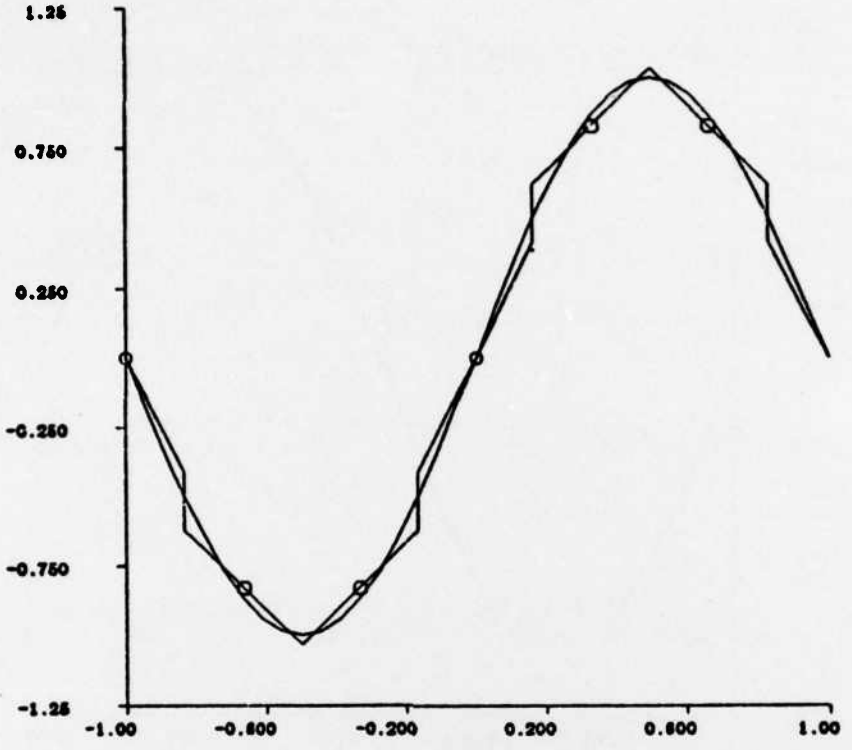


Fig. 1c. $P_I = 2$, $P_R = 1$.

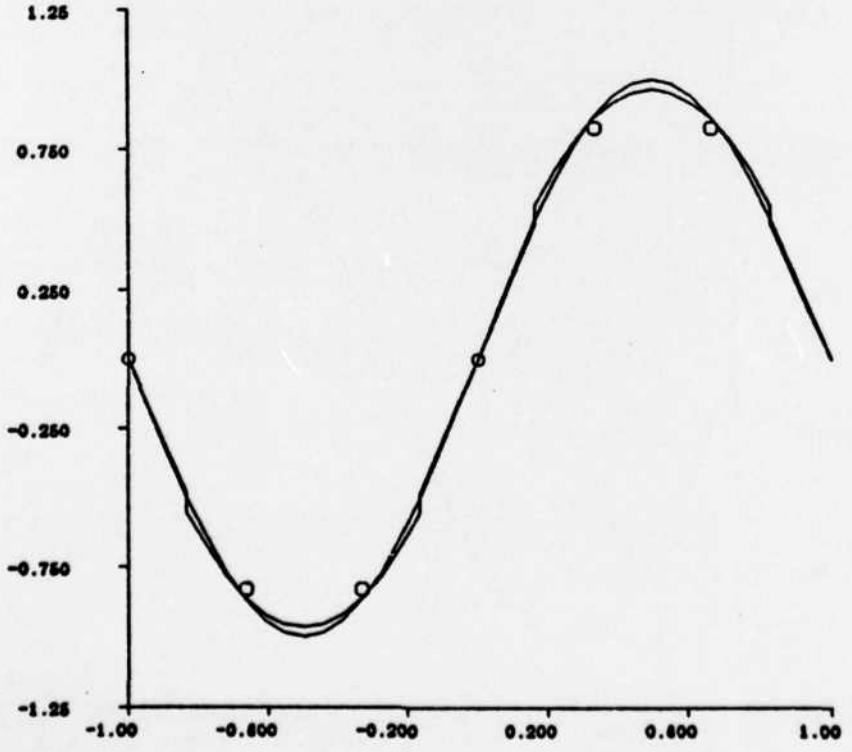


Fig. 1d. $P_I = 3$, $P_R = 2$.

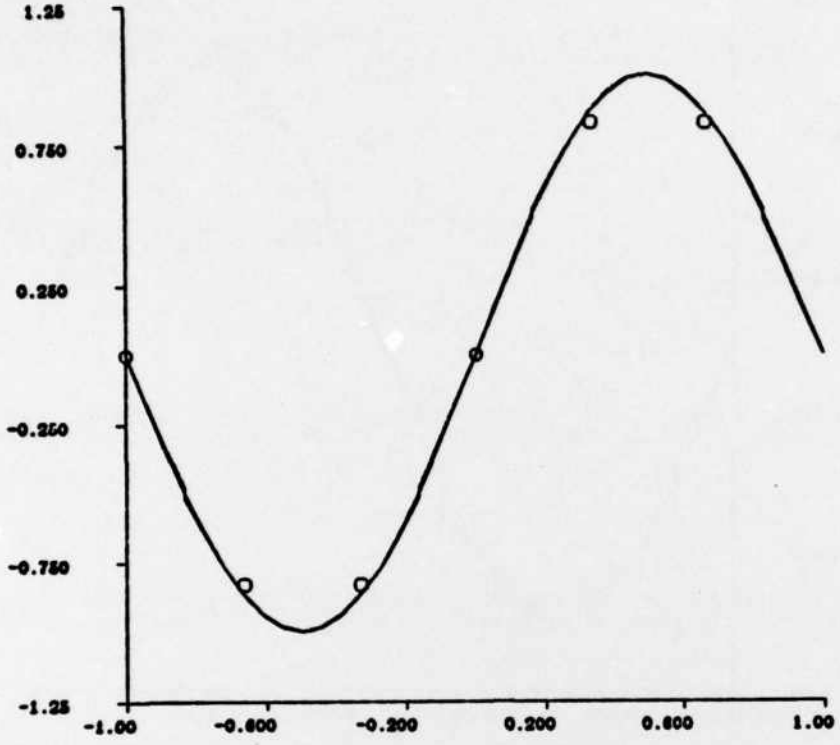


Fig. 1e. $P_I = 4$, $P_R = 3$.

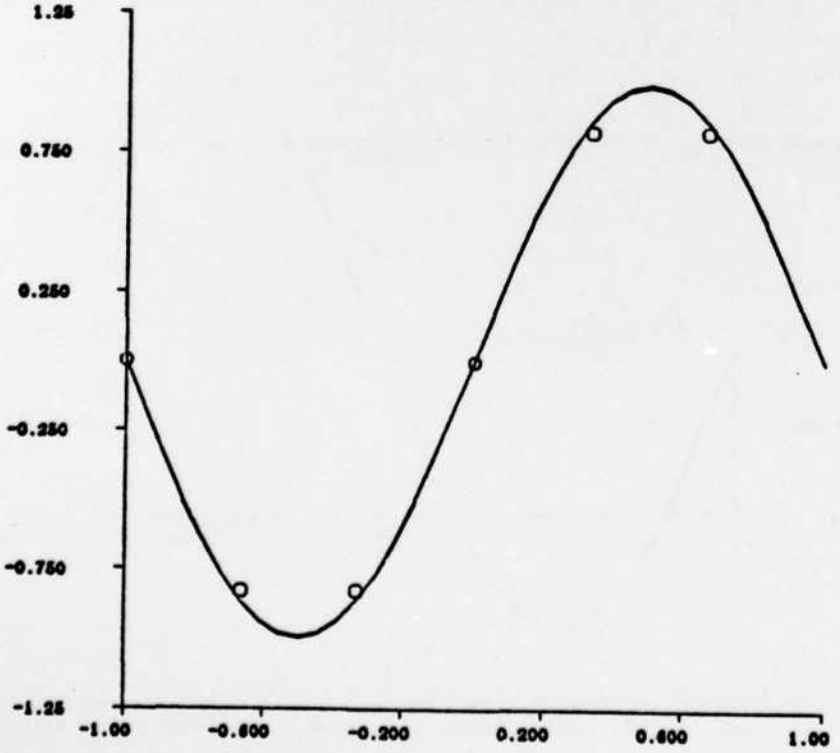


Fig. 1f. $P_I = 5$, $P_R = 4$.

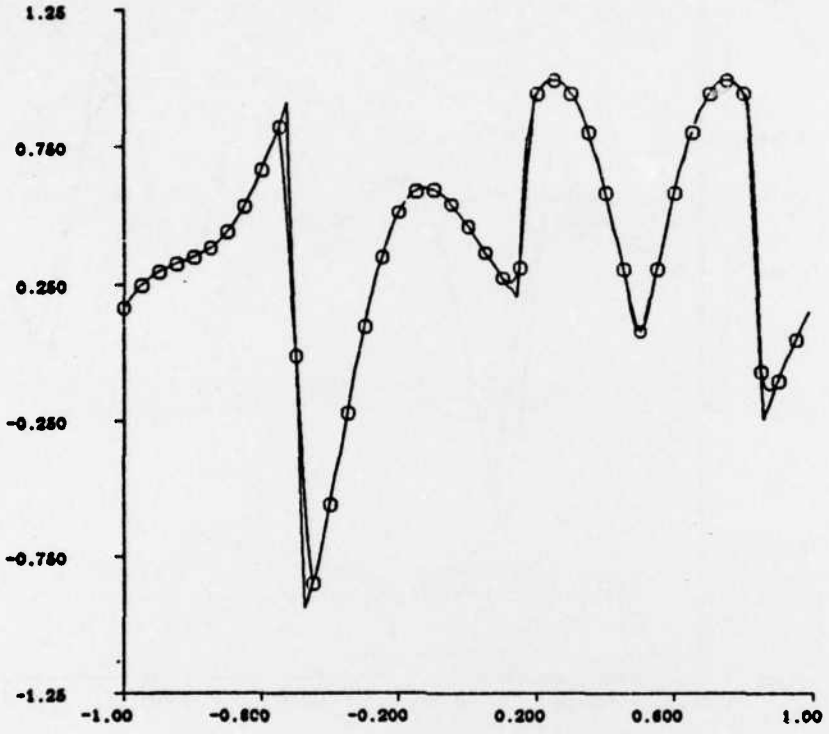


Fig. 2a. $H(x;\bar{u})$ with $P_I = 6$ vs. $\bar{u}(x)$. N = 40.

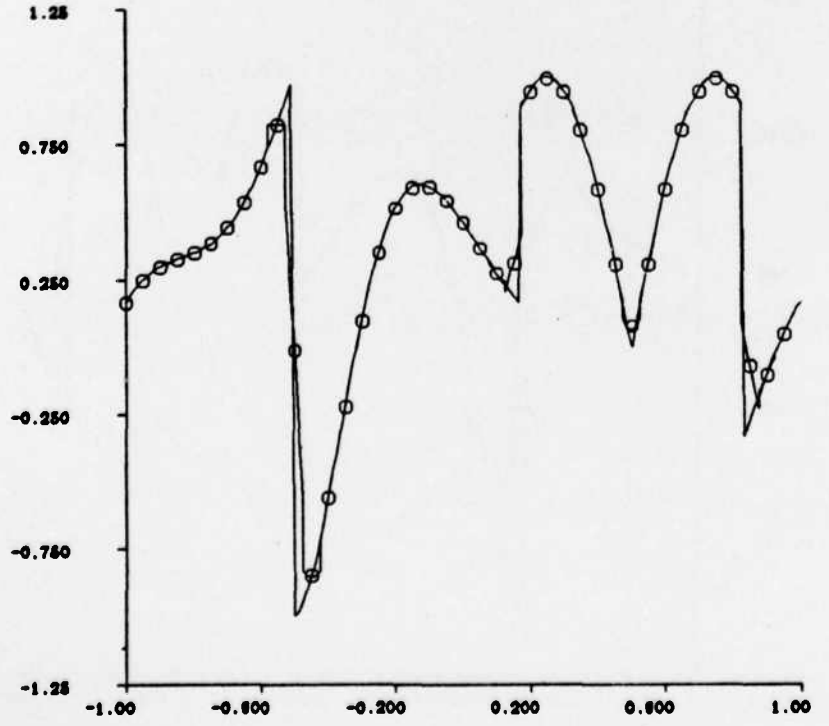


Fig. 2b. $R(x;\bar{u})$ with $P_I = 6$, $P_R = 5$ vs. $u(x)$. N = 40.

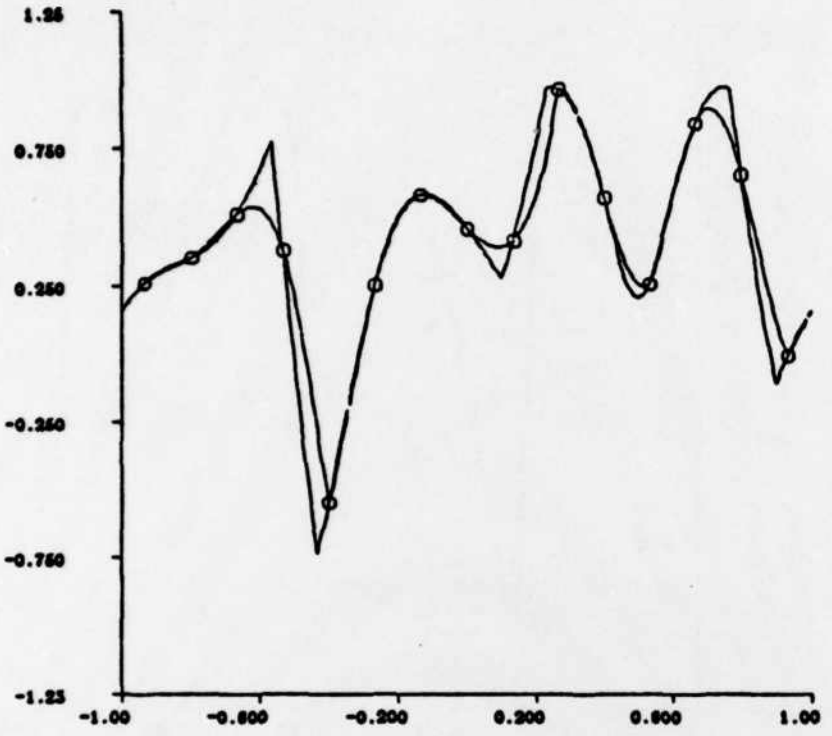


Fig. 3a. $H(x;\bar{u})$ with $P_I = 6$ vs. $\bar{u}(x)$. $N = 15$.

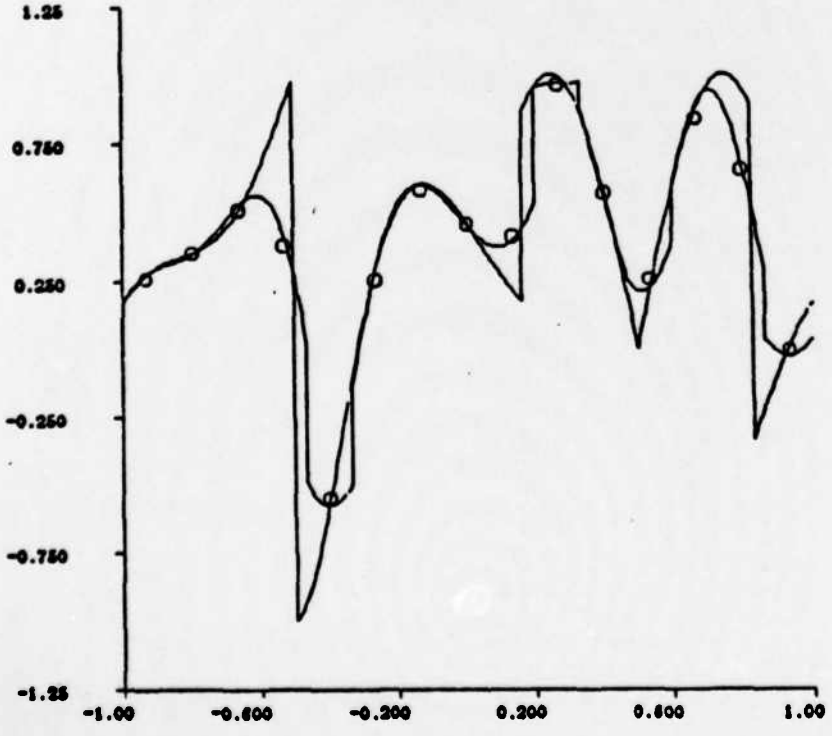


Fig. 3b. $R(x;\bar{u})$ with $P_I = 6$, $P_R = 5$ vs. $u(x)$. $N = 15$.

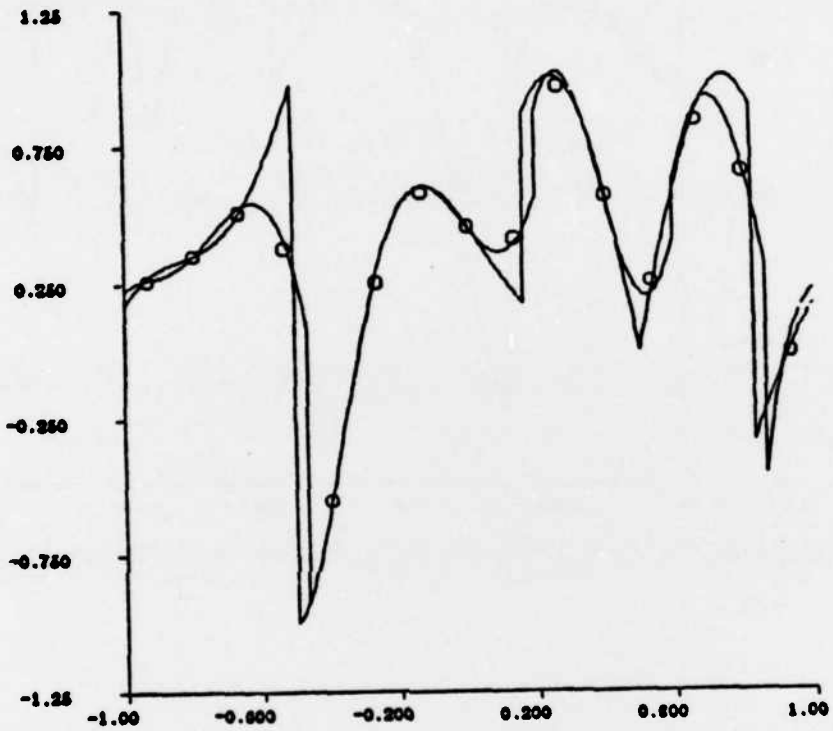


Fig. 3c. Reconstruction via Primitive function. $R(x;\bar{u})$ with $P_U = 6$ vs. $u(x)$. $N = 15$.

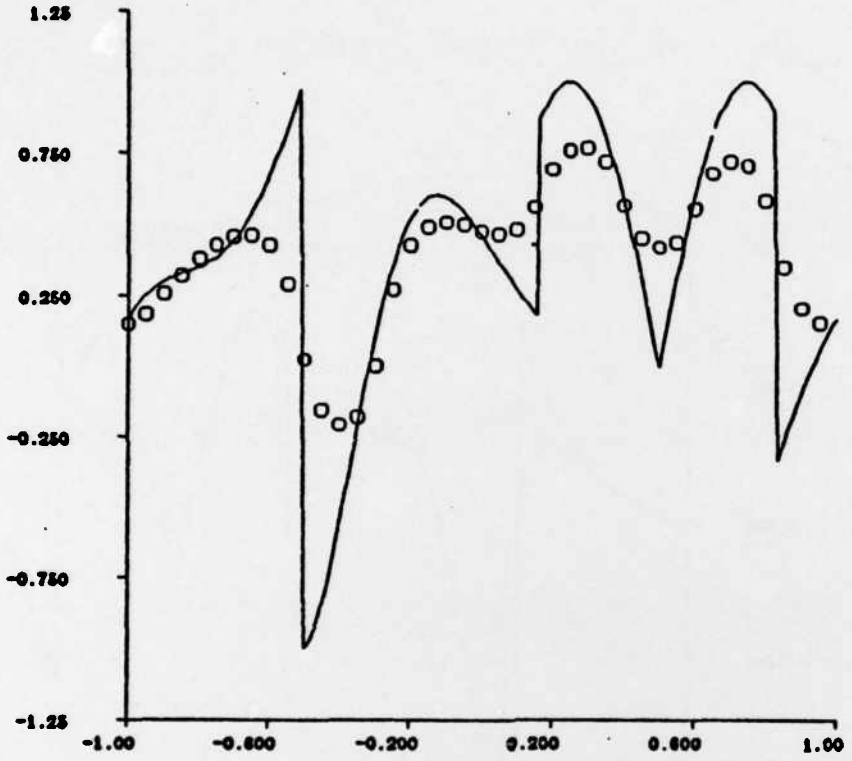


Fig. 4a. Solution with $P_I = P_R = 1$ and N = 40 at t = 2.

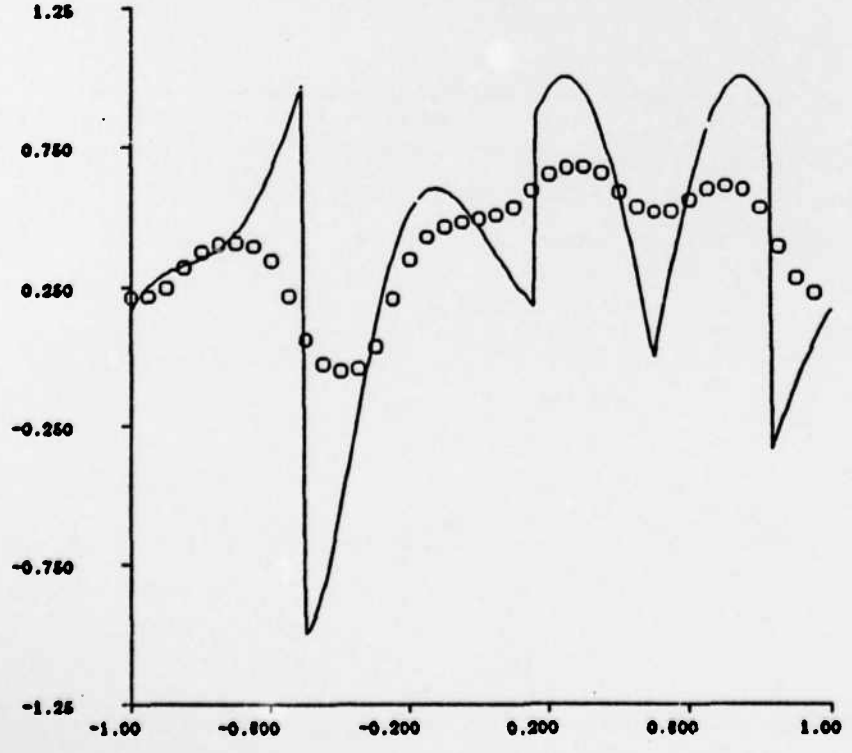


Fig. 4b. Same as Fig. 4a at t = 4.

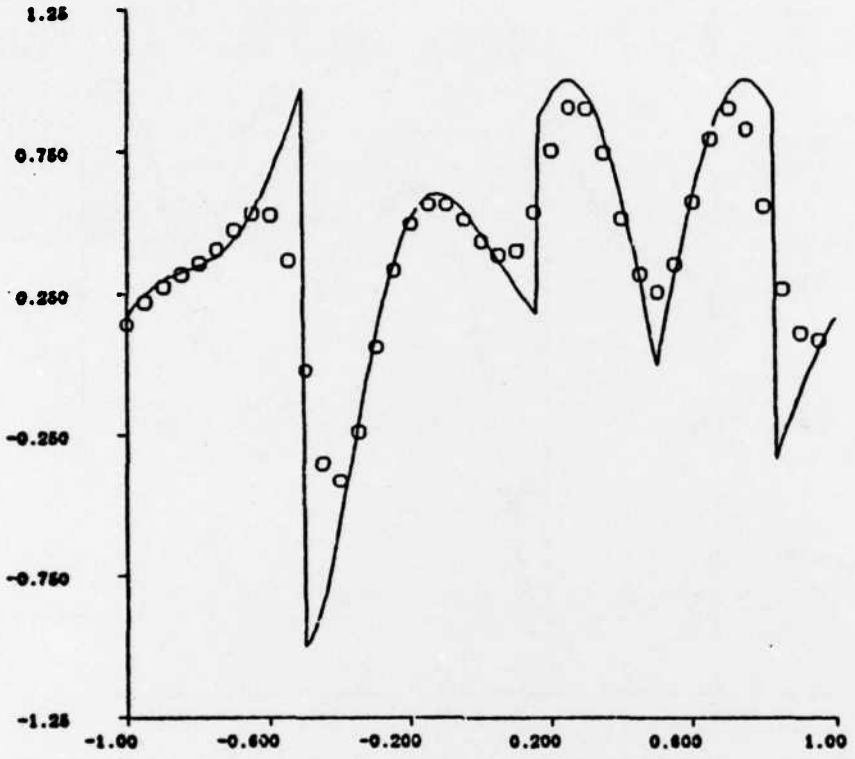


Fig. 5a. Solution with $P_I = 2$, $P_R = 1$ and $N = 40$ at $t = 2$.

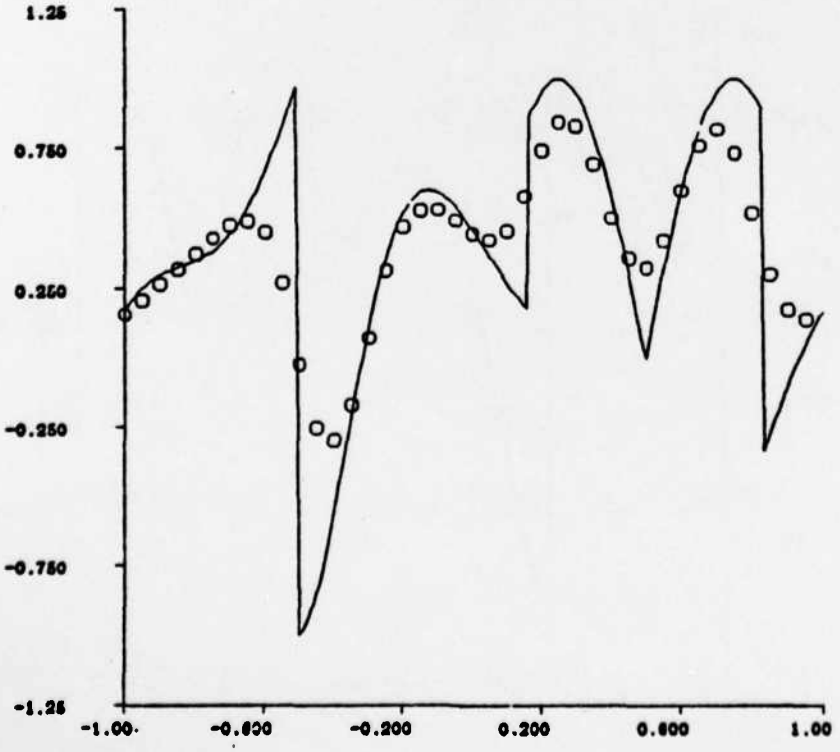


Fig. 5b. Same as Fig. 5a at $t = 4$.

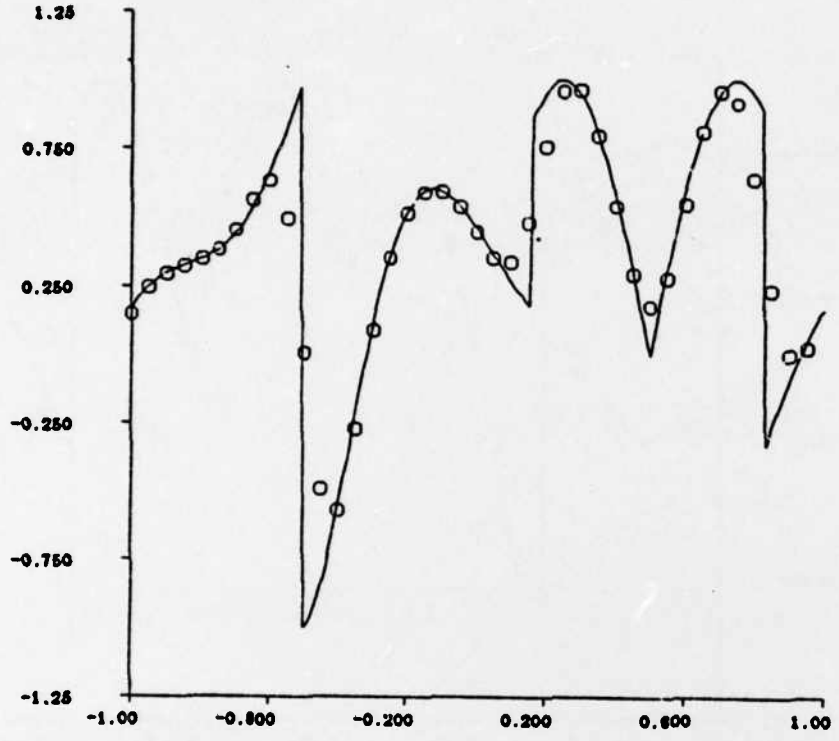


Fig. 6a. Solution with $P_I = 4$, $P_R = 3$ and $N = 40$ at $t = 2$.

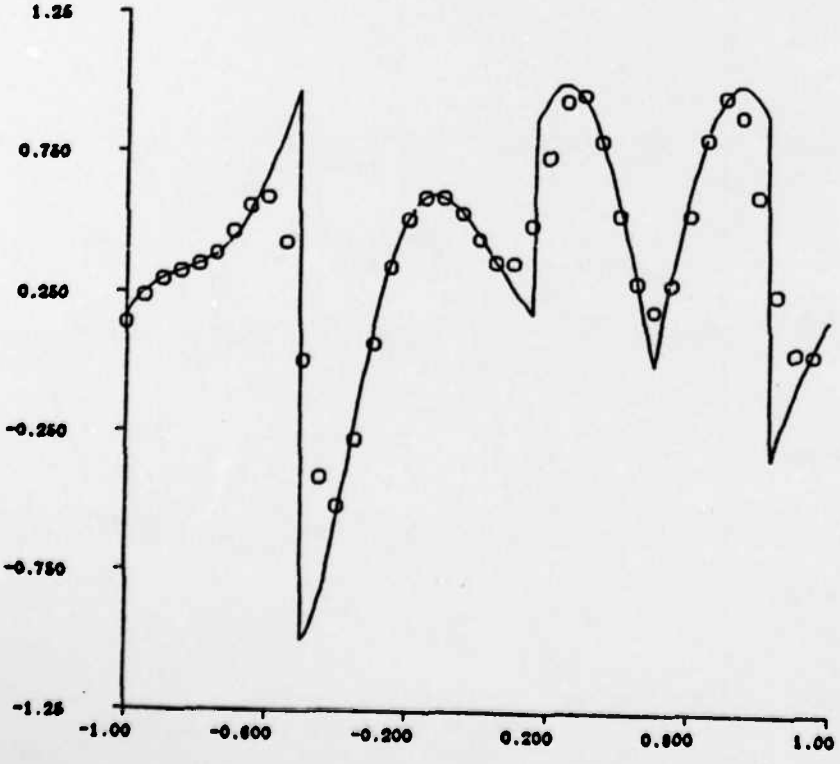


Fig. 6b. Same as Fig. 6a at $t = 4$.

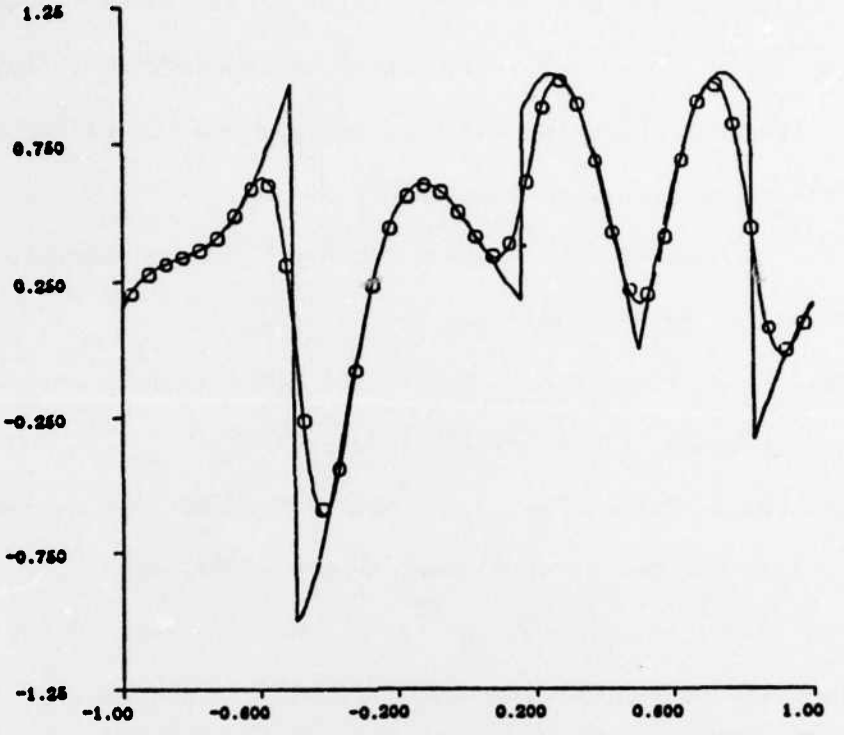


Fig. 7a. Solution using reconstruction via primitive function with $P_U = 6$ at $t = 2$.

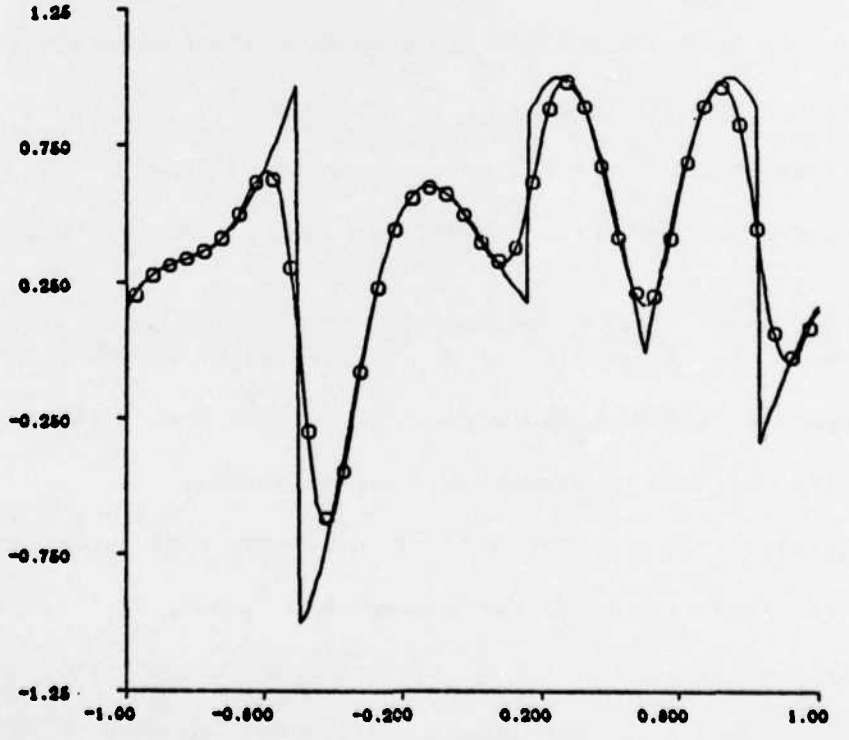


Fig. 7b. Solution using reconstruction via deconvolution with $P_I = 5$, $P_R = 4$ at $t = 2$.

References

[1] P. Colella and P. R. Woodward, The piecewise-parabolic method (PPM) for gas-dynamical simulations, J. Comp. Phys. v. 54, (1984), 174-201.

[2] A. Harten, J. M. Hyman and P. D. Lax (with appendix by B. Keyfitz), "On finite-difference approximations and entropy conditions for shocks", Comm. Pure Appl. Math., v. 29, (1976), 297-322.

[3] A. Harten, High resolution schemes for hyperbolic conservation laws, J. Comp. Phys., 49 (1983), 357-393.

[4] A. Harten, On a class of high resolution total-variation-stable finite-difference schemes, SINUM, v. 21, (1984), 1-23.

[5] A. Harten and S. Osher, "Uniformly high-order accurate non-oscillatory schemes, I.", MRC Technical Summary Report #2823, May 1985.

[6] A. Harten, S. Osher, B. Engquist and S. Chakravarthy, "Uniformly high-order accurate non-oscillatory schemes, II", in preparation.

[7] S. Osher and S. R. Chakravarthy, "High-resolution schemes and the entropy condition", SINUM, v. 21, (1984), 955-984.

[8] S. Osher and S. R. Chakravarthy, "Very high order accurate TVD schemes", ICASE Report #84-44, (1984).

[9] B. Van Leer, Towards the ultimate conservative scheme, II. Monotonicity and conservation combined in a second order scheme, J. Comp. Phys. 14 (1974), 361-376.

[10] P. Woodward, in Proceedings of the NATO Advanced Workshop in Astrophysical Radiation Hydrodynamics, Munich, West Germany, August 1982; also Lawrence Livermore Lab. Report #90009.

[11] S. T. Zalesak, "Very high-order and pseudo-spectral flux-corrected transport (FCT) algorithms for conservation laws", in "Advances in computer methods for partial differential equations", Vol. 4 (R. Vichnevetsky and R. S. Stepleman, eds.) IMACS, Rutgers University, 1981.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER #2829	2. GOVT ACCESSION NO. AD-A158 131	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) On High-Order Accurate Interpolation for Non-Oscillatory Shock Capturing Schemes		5. TYPE OF REPORT & PERIOD COVERED Summary Report - no specific reporting period 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Ami Harten		8. CONTRACT OR GRANT NUMBER(s) DAAG29-82-0090 DAAG29-80-C-0041 NCA2-IR390-403
9. PERFORMING ORGANIZATION NAME AND ADDRESS Mathematics Research Center, University of Wisconsin 610 Walnut Street Madison, Wisconsin 53705		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Work Unit Number 3 - Numerical Analysis and Scientific Computing
11. CONTROLLING OFFICE NAME AND ADDRESS See Item 18 below		12. REPORT DATE June 1985 13. NUMBER OF PAGES 40
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

U. S. Army Research Office
P. O. Box 12211
Research Triangle Park
North Carolina 27709

National Aeronautics and Space Administration
Washington, DC 20546

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Conservation laws, Godunov-type schemes, high-order accuracy, non-oscillatory interpolation, Gibbs-phenomenon, cell-average.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

In this paper we describe high-order accurate Godunov-type schemes for the computation of weak solutions of hyperbolic conservation laws that are essentially non-oscillatory. We show that the problem of designing such schemes reduces to a problem in approximation of functions, namely that of reconstructing a piecewise smooth function from its given cell averages to high order accuracy and without introducing large spurious oscillations. To solve this reconstruction problem we introduce a new interpolation technique that when applied to piecewise smooth data gives high-order accuracy wherever the function is smooth but avoids having a Gibbs-phenomenon at discontinuities.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

END

FILMED

9-85

DTIC